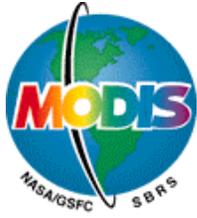




MODIS Science Software and Product Versioning

Robert Wolfe, MODLAND Support
and
Mike Linda, SDST

MODIS Programmers Forum Breakout Session
MODIS Science Team Meeting
December 15, 1998



Introduction

- Versioning currently is *Ad Hoc*
- Need for formal convention recognized by Science Team
- White paper on versioning released
 - Discusses versioning framework science software and products
 - Input from the Science Team members and developers requested
- Implementation details and impact still need to be addressed

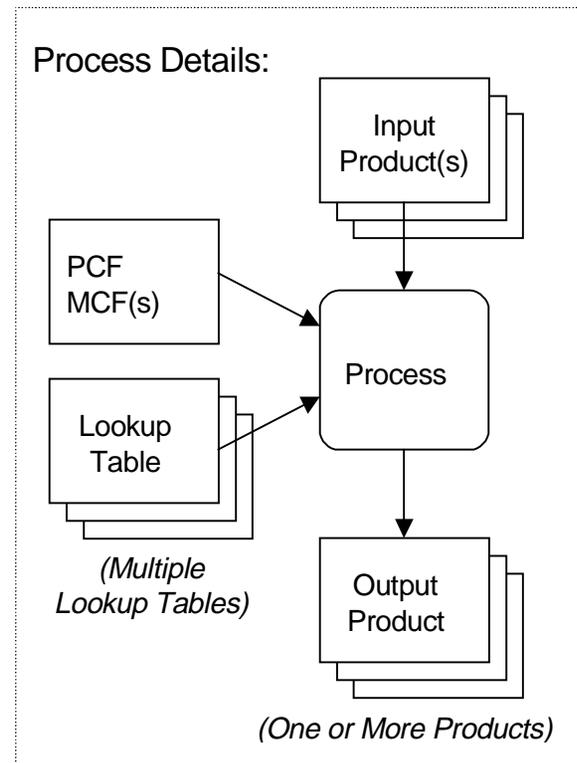
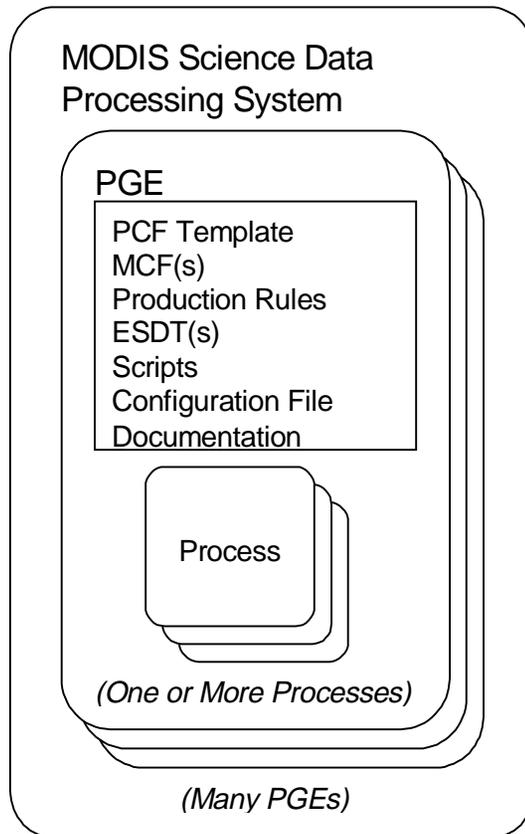


Requirements for Versioning

1. Easy to understand
2. Provides a unique label for elements in the system that may change
3. Conveys information about the significance of a science algorithm change
4. Conveys information about the version of the system



Where Versioning is Used





Versioning Summary

Area	Format	Example	Location in Output Product	Where Stored
System	<i>major.minor</i>	2.1	external to product; table of individual elements' versions	SDST CM
Process	<i>major.minor.update</i>	2.1.8	<i>ProcessVersion</i> - Inventory Metadata Product Specific Attribute (PSA)	Process Include File
PGE	<i>major.minor.update</i>	2.3.1	<i>PGEVersion</i> - Inventory Metadata Attribute	PCF Template
ESDT	integer from 0 to 255	2	<i>VersionID</i> - Inventory Metadata Attribute	MCF
Lookup Table	free form	2.3	none specified, possibly as part of file name in <i>InputPointer</i> Inventory Metadata Attribute	File Name or Internal to the Table
Product	<i>major.minor.update</i>	2.1.3	<i>LocalVersionID</i> - Inventory Metadata Attribute and as Part of Local Granule ID	PCF Template



System Version

- Major and minor version numbers - example: 2.1
- Labels a “virtual” build of the system
 - Changes when one or more PGEs becomes incompatible with a given system
- Major version - signifies a major project milestone
 - Controlled by Science Team leader
 - e.g., 3.x is the first major reprocessing
- Minor version - documents a unique internally-compatible version
 - Controlled by SDST and science team
 - Documented in system description document
 - Change interval: 3-12 months
- One-way mapping from system version to versions of elements
- Not stored in product



Process Version

- Major, minor and update version numbers - example: 2.1.3
- Labels science software
- Major version - signifies a major project milestone
 - Changes when significant changes to process are made near major project milestone
 - Synchronizes processes with milestones (only if process changes)
- Minor version - signifies a change to the science algorithm(s)
 - Controlled by science team member
 - Large number of changes are discouraged to prevent end-user confusion
- Update version - any other changes
 - Error message changes
 - Minor changes to product format
- Updates to software libraries (or the like) are handled on a case-by-case basis
- Product location: *ProcessVersion* ECS Inventory Metadata PSA
- Likely be stored in include file, e.g., “version.h”
- Some scientists (oceans) may also identify each algorithm in product uniquely using *VerParm_**



PGE Version

- Major, minor and update version numbers - example: 2.1.3
- Labels all items tied to a PGE:
 - MCFs, PCF templates, etc.
 - But not processes
- Loosely tied to version of its processes
 - Major/minor/update version number changes when corresponding process version changes
 - Not the same because PGE can contain multiple processes
 - Updates to other (non-process) items usually change the update version number
- Product location: *PGEVersion* ECS Inventory Metadata
- Stored in PCF template - runtime parameter
 - Single LUN for all PGEs



ESDT

- CM Version - form unknown
 - Internally tracks specific set of ECS files related to the ESDT
 - Used by the DAAC for configuration control
- Public Version - integer from 0 to 255
 - Distinguishes different sub-types of ESDTs
 - Changes when any inventory metadata field is deleted or when a mandatory metadata field is added
 - Other changes controlled by the SCF
 - Number of changes should be minimized
- Public version affect product ordering via ECS interface
 - Must select ESDT/version before selecting other search criteria
 - Future enhancements (when?) will allow searching across multiple versions
- Public version stored in MCF and written to *VersionID* ECS Inventory Metadata



Lookup Table

- Free form format
 - Use by the SCF
 - May be part of file name or internal to file
- File name:
 - Clearly delineated field - like: “lookuptable.v12.dat” is the 12th version
- Internal: Text vs. Binary
 - Clearly documented in either case
 - Binary - should be accessible through “string | grep” commands
 - Text - should be easily identifiable
- Should be some information in each product which identifies which version was used in generating the product
- Unified policy is not established
 - Strongly urge obvious versioning: version number in filename
 - Discourage hidden versions (binary implementation internal to file)



Product Version

- Should identify significant changes in the product science content
 - SCF determines "significant"
 - Not a problem if the call is false
- Major, minor and update version numbers - example: 2.1.3
- Similar to process version
 - Major - resynchronize at major milestones
 - Minor - change in science content
 - Update - any other changes
- May include
 - Software/algorithm changes
 - Changes to the production rules
 - Changes to lookup tables
- Product location
 - *LocalVersionID* ECS Inventory Metadata
 - Part of *LocalGranuleID* - "v" prefix; "." replaced with "_"; e.g., v2_1_3
- Stored in PCF template - runtime parameter; unique for each product



Product History String

- Concatenation of product history strings of all input products
 - Duplicates removed
 - Only highest version kept for each input ESDT
 - Current product version added to front
 - e.g., “MOD09:2.2.3 MOD35_L2:2.1.0
MOD02QKM:2.2.3 ...”
- Helps to identify some (not all) changes in upstream products
- Product location: *ProductionHistory* ECS Archive Metadata



Implementation

- Most scientists will implement using current internal tools/techniques
- SDST may develop modify common library routines for
 - Local granule ID
 - Product history
- Phase-in planned
 - Expected in any new or updated code deliveries
- SDST will
 - register metadata changes with ECS
 - coordinate LUN assignments
 - will maintain a versioning web page reference for all



Issues

- Some areas are not fully defined
 - ESDTs
 - Lookup tables
- Still complicated (see # 1 in the requirements)
- Need approval/consensus on this framework to move forward
 - Any major problems?
 - Need something in place before launch
 - Will be useful in pre-launch testing
- Discussion